RATE-ACCURACY TRADE-OFF IN VIDEO CLASSIFICATION WITH DEEP CONVOLUTIONAL NEURAL NETWORKS

Alhabib Abbas†, *Mohammad Jubran*‡, *Aaron Chadha*†, *Yiannis Andreopoulos*†

† Dept. of Electronic & Electrical Engineering, University College London, London, U.K.
‡ Dept. of Electrical & Computer Engineering, Birzeit University, West Bank, Palestine

ABSTRACT

Advanced video classification systems decode video frames to derive the necessary texture and motion representations for ingestion and analysis by spatio-temporal deep convolutional neural networks (CNNs). However, when considering visual Internet-of-Things applications, surveillance systems and semantic crawlers of large video repositories, the compressed video content and the CNN-based semantic analysis parts do not tend to be co-located. This necessitates the transport of compressed video over networks and incurs significant overhead in bandwidth and energy consumption, thereby significantly undermining the deployment potential of such systems. In this paper, we investigate the trade-off between the encoding bitrate and the achievable accuracy of CNN-based video classification that ingests AVC/H.264 encoded videos. Instead of entire compressed video bitstreams, we only retain motion vector and selected texture information at significantly reduced bitrates. Based on two CNN architectures and two action recognition datasets, we achieve 38%–59% saving in bitrate with marginal impact in classification accuracy. A simple rate-based selection between the two CNNs shows that even further bitrate savings are possible with graceful degradation in accuracy. This may allow for rate/accuracy-optimized CNN-based video classification over networks.

Index Terms— Video classification, convolutional neural networks, video streaming

1. INTRODUCTION

Action or event recognition and video classification for visual Internet of Things (IoT) systems [1], video surveillance [2], and fast analysis of large-scale video libraries [3] have been advancing rapidly due to the advent of deep convolutional neural networks (CNNs). Given that such CNNs are very computationally and memory intensive, they are not commonly deployed at the video sensing/parsing nodes of the system (a.k.a., "edge" nodes). Instead, video is either transported to certain high-performance aggregator nodes in the network [1] that carry out the CNN-based processing, or compact features are precomputed in order to allow for less complex on-board processing at the edge [3], at the expense of some accuracy loss for the classification or recognition task.

Motion vector based optical flow approximations have been proposed for action recognition by Kantorov and Laptev [4], albeit without the use of CNNs. In more recent work, proposals have been put forward for fast video classification based on CNNs that ingest compressed-domain motion vectors and selective RGB texture information [5, 6]. Despite their significant speed and accuracy improvements, none of these approaches considered the trade-off between rate and classification accuracy obtained from a CNN. Conversely, while rate-accuracy trade-offs have been analysed for conventional image and video feature extraction systems [7, 8], these studies do not cover deep CNNs and semantic video classification, where the different nature of the spatio-temporal classifiers can lead to different rate-accuracy trade-offs.

In this paper, we show that crawling and classification of remote video data lakes can be achieved with significantly-reduced bandwidth requirements by exploring the rate-accuracy trade-off in CNN-based video classification. Specifically, we show how to reduce the compressed bitstream elements needed for two-stream CNN classification with minimal modifications in the required syntax (with emphasis on the H.264/AVC standard). Such selectively cropping of the required texture and motion vector elements of a compressed video bitstream is referred to as a *cropped bitstream*. We show that the video classification accuracy can be tuned according to the bitrate required to stream the cropped bitstream to the utilized CNN. An interesting observation from our results is that, unlike rate-distortion, rate-accuracy can be not monotonic for CNN-based classification. Finally, we outline some interesting avenues for future research investigation.

We acknowledge support from: EPSRC (projects EP/P02243X/1 and EP/R025290/1), Innovate UK (project DELVE-VIDEO 132739), the Leverhulme Trust (RAEng/Leverhulme Senior Research Fellowship of Y. Andreopoulos), the Royal Commission for the Exhibition of 1851 (Fellowship of A. Chadha) and EPSRC CASE (PhD studentship of A. Chadha, co-sponsored by BAFTA). M. Jubran performed the work while visiting University College London under a "Distinguished Scholar Award" from the Arab Fund Fellowships programme.

2. CROPPED VIDEO BITSTREAMS

We base our cropped bitstream engine on the JM implementation of the ITU-T & ISO/IEC H.264/AVC standard[1] [9, 10]. The aim is to reduce the bitrate of the compressed bitstream while preserving the information contained within key texture components and motion vectors that are of paramount importance for semantic video classification. In H.264/AVC, pictures are split into 16×16 pixel macroblocks (MB) to represent luminance samples, with two corresponding 8×8 chroma blocks (for 4:2:0 chroma subsampling). Macroblocks are the core of the coding layer and form the basis for the adaptive inter and intra predictions. Each of the inter-predicted macroblocks is encoded using blocks from the set $\{16 \times 16, \dot{1}6 \times 8, \dot{8} \times 16, \dot{8} \times 8\}$ [10, 11]. Blocks are predicted via translational motion vectors (MVs) that represent the displacement from matching blocks in previous or subsequent reference frames. Increasing the number of small-size blocks increases the granularity of the MV grid at the expense of lower coding efficiency. These MVs represent the temporal activity and have been shown to be highly correlated with optical flow estimates [5]. If the area covered by the MB is static, the MB is "skipped' and is not encoded. The resulting prediction residual from temporal prediction of non-skipped MBs is encoded using transform coding. The transform coefficients are then quantized based on the quantization parameter (QP). The QP value per frame can be chosen from 52 values in $[0, 51]$ [10], with lower values indicating high-fidelity encoding.

In our work, only selected subsets of the quantized transform coefficients will be entropy encoded and then included in the cropped bitstream. This set of coefficients is the information transmitted to the CNN to classify the spatial activity of the sensor. The first frame of every video sequence is encoded as an Instantaneous Decoding Refresh (IDR) and all subsequent frames in the video are encoded as uni-directionally predicted (P) or intra-predicted (I) frames.

2.1. Selective Texture and MV Information

In order to reduce the bitrate of the bitstream, we can employ selective retention of texture information by retaining the texture information of active regions [5]. To implement selective writing in the AVC reference software (JM 19.0), we modified the `writeCoeff4x4_CAVLC_normal()` and `write_chroma_intra_pred_mode()` in `macroblock.c` at the encoder and the `read_coeff_4x4_CAVLC()` in `read_comp_cavlc.c` and `read_ipred_modes()` in `mb_read.c` at the decoder to allow for a skip symbol for all non-active areas [9]. To simplify our tests, we retain the texture of the IDR frame and skip all texture of P-frames with a single skip symbol. Moreover, to derive a temporal activity map from the P-frame MVs, we apply the following steps: *(i)* MVs are extracted from the compressed bitstream using the `read_motion_info_from_NAL_p_slice()` function in `macroblock.c` of JM 19.0 and mapped to an 8×8 grid, wherein each point in the grid is set to be 8 pixels away from its neighboring points; *(ii)* MVs are interpolated wherever a macroblock does not contain a motion vector but two or more of its neighbors do. All other syntax elements (including modes and motion information) are left as specified in the H.264/AVC standard.

[1] due to the current popularity of H.264/AVC content, we shall be focusing on this standard in this paper. Future work will validate our approach with the more recent HEVC standard.

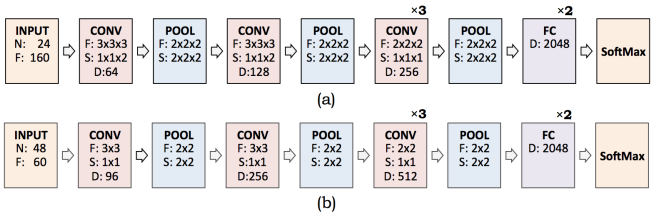

Fig. 1: Utilized CNNs: (a) 3D temporal CNN architecture; (b) 2D temporal CNN architecture. (N: cropped input size F: reception field size S: convolution stride D: filter depth)

3. PROPOSED FRAMEWORK FOR COMPRESSED-DOMAIN CLASSIFICATION

3.1. CNN Architectures

In Fig. 1 we illustrate the two CNNs used for the temporal stream to produce our results. We use two architectures in order to demonstrate that our rate-accuracy CNN-based classification is applicable with different deepnet architectures.

The first CNN architecture we consider is the 3D CNN proposed by Chadha *et al.* [5]. As illustrated in Fig. 1.a, all convolutional and pooling layers are spatiotemporal in extent, which subsequently enables the model to capture the motion information between consecutive motion vector frames. Crucially, the spatiotemporal features are expected to improve classification performance between similar actions. We generate a 4D motion vector input by splitting the dx and dy vector components into separate channels, thus resulting in a $W \times H \times 2 \times F$ volume. We compensate for the low resolution of the extracted motion vector frames by setting a long temporal extent F of 160, which typically comprises the entire video duration.

The second architecture we consider is a 2D CNN, as illustrated in Fig. 1.b. The model design is based off ClarifaiNet [12] and only comprises 2D spatial filters; we notably reduce the size of the first filter from 7×7 to 3×3 and decrease the stride of the first two convolutional layers to 1×1. A similar architecture was also employed in recent work on fast video classification [6]. The input is generated by stacking the motion vector dx and dy components into a single $W \times H \times 2F$ volume, where the temporal depth F is set as

60. In general, 2D CNNs are able to achieve a lower complexity than 3D CNNs, whilst forgoing modelling any temporal dependencies. Nonetheless, the lower complexity means we can afford to use a higher input spatial resolution. We use bilinear interpolation to double the size of the spatial input, which enables the 2D filters to learn more distinguishing spatial features.

Finally, concerning spatial processing of RGB texture, we use the well-established VGG-16 [13] CNN architecture to classify RGB frames and capture motion-invariant spatial features of video content. Our spatial CNN is pre-trained on ImageNet [14] and fine-tuned on UCF-101. The spatial stream ingests the decoded frames per video and the predictions made by the spatial CNN are ultimately fused with the predictions from the temporal stream to produce the final two-stream classifier decisions.

3.2. Training and Testing

We train both temporal stream architectures using stochastic gradient descent with momentum set to 0.9. The initialization of He *et al.* [15] is used and weights are initialized from a normal distribution. Mini-batches of size 64 are generated by randomly selecting 64 training videos per batch. The learning rate is initially set to 10^{-2} and is decreased by a factor of 0.1 every 30k iterations. The training is completed after 90k iterations.

We follow the heavy data augmentation practices utilized in recent work [5], in order to minimize overfitting for both the 2D and 3D CNN. These include a multi-scale random cropping of the input and spatial resizing to a fixed size N, followed by zero centering the motion vector field by subtracting the mean motion vector from the volume. For the 3D CNN, the fixed crop size is set to 24, whereas for the 2D CNN this is doubled to 48. In addition, we use a dropout ratio to 0.5 for the first two fully connected layers in both models. During testing, for the temporal stream we generate 10 random volumes of temporal size F from which to test on. Per volume, we use the standard 10-crop testing, cropping the four corners and the center of the image to spatial size $N \times N$ and considering both horizontally flipped and unflipped versions. As such, we average the scores over 10 crops and 10 volumes to produce a single score for the video. For the spatial stream, we use the single IDR frame of each video and extract multiple crops by flipping and resizing to the input size of the VGG-16.

3.3. Dual Temporal CNN Classifier

Since $F_{3D} > F_{2D}$, our 3D-CNN requires more frames than the 2D-CNN in order to derive a classification result. In addition, we expect that denser MV frames will benefit from the larger spatial dimensions of the 2D-CNN. Since the density of inputs to the temporal stream is proportional to the MV bitrate, R_{motion}, we use our 3D CNN for all videos with $R_{\text{motion}} < \lambda$ and the 2D CNN for the remaining set of videos, with λ a rate-accuracy optimization parameter whose value can be tuned to fit operational conditions as we show in Section 4.3. We remark that, while in this paper the value of R_{motion} is derived experimentally during the encoding of each video, for offline rate-accuracy optimization studies it can also be derived via rate-distortion models, e.g., via well-established rate-distortion models for H.264/AVC [16].

4. EXPERIMENTAL RESULTS

4.1. Used Datasets

We train and test our 2D and 3D CNN architectures on four distinct motion vector datasets generated by varying the QP of H.264 to encode UCF-101 [17] while skipping texture information as described in Section 2. For all videos: the first frame is encoded as an IDR (with I period set to 100 frames), the frame rate is set to 25, and we set the motion vector search range to 16 pixels. Since specifying a particular quantization parameter has a direct effect on the MVs produced by H.264, this gives several distinct source distributions for the classifier to be trained and tested on. Table 1 reports results comparing the original bitrate, R_{orig}, with the bitrate of the cropped bitstreams, R_{cropped}, with R_{motion} kbps comprising the MV information. The experiments with UCF-101 show that streaming cropped bitstreams allows for 28% to 92.5% reduction in bitrate (R_{cropped} vs. R_{orig} for QP=51 down to QP=0), with 23% to 48% of the cropped bitstream comprising MV information. Importantly, for QP values of $[30, 40]$, our framework allows for 38%–59% saving in bitrate with the impact in the corresponding classification accuracy shown to be marginal, as discussed in the following section.

4.2. Rate-Accuracy Results

As the quality of predictions made by deep CNNs are strongly tied to properties of the source distribution (e.g. cross-class variance, noise), we expect that varying the rate should affect on the accuracy of our classifier. The related results are presented in Fig. 2. It is interesting to note that the utilized CNNs exhibit their best accuracies around QP values of $[30, 40]$ and the rate-accuracy curves are not monotonic, i.e., accuracy decreases for very low or very high QP values. We expect sparser motion vectors to make certain classes with high motion similarity particulary harder to classify and easier to confuse with each other (e.g., inputs produced by setting QP $= 51$). On the other hand, we have observed that decreasing QP below 30 also has a detrimental effect to accuracy, since the derived MVs become significantly more noisy due to the inadequacy of the simple translational block model of H.264/AVC to smoothly approximate the true motion field in the video scenes [5, 18]. To illustrate this further, Fig. 2 includes results when restricting the H.264/AVC temporal pre-

Table 1: Average bitrate (kbps) of texture and motion information for the UCF-101 data set for the original bitstream, R_{orig}, and the cropped bitstream, R_{cropped}.

QP	R_{orig}	R_{cropped}	R_{motion}	% of R_{motion} to R_{orig}	% of R_{motion} to R_{cropped}
0	4273.0	321.3	155.4	3.6	48.3
30	274.9	112.3	46.9	17.0	41.7
40	80.0	49.9	18.5	23.2	37.1
51	27.7	20.0	4.6	16.7	23.1

diction to 16×16 blocks. As expected, these cases show higher accuracy loss and variation for very low and very high QP values since the translational block motion model fails to provide an accurate representation for these extreme cases.

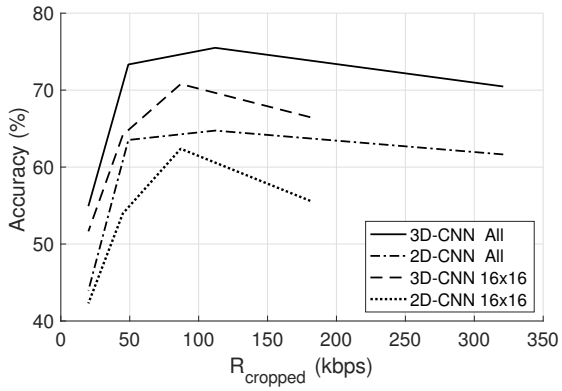

Fig. 2: Rate-accuracy after cropped bitstreams are passed to the 2D and 3D temporal CNNs. Each point for every curve corresponds to a different QP setting during encoding, with "16×16" indicating the restriction to 16×16 blocks (no MB subblocks) and "All" indicating the use of all MB partitions.

In Table 2. we report the results for our two-stream classifier (fusion of predictions of temporal and spatial CNNs) with the two best quantization settings from Fig. 2 and all block sizes. Our results show that our approach remains competitive to the state-of-the-art for UCF-101, while retaining the significant bitrate gains reported in Table 1. In addition, while our approach is outperformed by TSCNN and LTC on HMDB, these methods are orders of magnitude more complex than operating with compressed-domain information [5, 6], since they require the use of optical flow or complex SVM fusion networks and need to decode the entire video bitstream.

4.3. Rate-Accuracy Classifier Selection and Future Work

Fig. 3 shows the potential for rate-accuracy based CNN selection based on the design of Section 3.3. By varying the threshold λ from inf to 30, we see that an additional 9 kbps reduction (further 18% reduction of R_{cropped}) can be made

Framework	Accuracy (%) UCF	HMDB
Proposed, QP $= 40$	88.1	48.2
Proposed, QP $= 51$	84.0	47.0
EMV + RGB-CNN [6]	86.4	–
MVCNN [5]	89.8	56.0
TSCNN (SVM fusion) [19]	88.0	59.4
LTC [20]	91.7	64.8
C3D (3 nets)+IDT [21]	90.4	–

Table 2: Comparison with state-of-the-art CNNs.

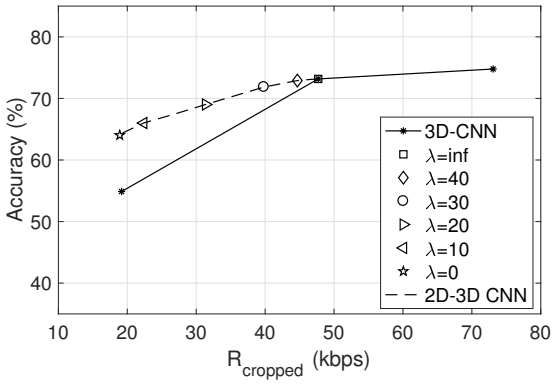

Fig. 3: Rate-accuracy using the setup of Section 3.3. The dashed line shows the accuracy when varying λ for QP=40 and selecting between the 2D and 3D CNN, while the solid line shows the rate-accuracy of the 3D CNN with varying QP.

at less than 2% reduction in classification accuracy. Importantly, even further bitrate reductions are made possible with graceful degradation in classification accuracy (lower values of λ). This shows the potential for further exploration of rate-accuracy optimization for CNN-based video classification.

5. CONCLUSION

We present the first exploration of rate-accuracy trade-offs in advanced video classification with CNNs. The obtained results show that 38%–59% saving in bitrate can be achieved when cropping compressed video bitstreams to the necessary elements for 2D or 3D CNN classification. In addition, we have observed that non-monotonic rate-accuracy curves are obtained when using such CNNs and H.264/AVC motion vectors. Finally, a rate-based selection between the 2D and 3D CNNs is shown to yield even further rate gains with minimal impact in classification accuracy. Further work may find further features that can be included in rate-accuracy optimization for advanced video classifiers within visual IoT or semantic video crawling applications.

6. REFERENCES

[1] Jorge Posada, Carlos Toro, Iñigo Barandiaran, David Oyarzun, Didier Stricker, Raffaele de Amicis, Eduardo B Pinto, Peter Eisert, Jürgen Döllner, and Ivan Vallarino, "Visual computing as a key enabling technology for industrie 4.0 and industrial internet," *IEEE Computer Graphics and Applications*, vol. 35, no. 2, pp. 26–40, 2015.

[2] Gang Yu and Junsong Yuan, "Fast action proposals for human action detection and search," in *Proc. IEEE Conf. Comput. Vis. Patt. Recog., CVPR*, 2015, pp. 1302–1311.

[3] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," *arXiv preprint arXiv:1609.08675*, 2016.

[4] Vadim Kantorov and Ivan Laptev, "Efficient feature extraction, encoding and classification for action recognition," in *Proc. IEEE Conf. Comp. Vis. Pattern Rec. (CVPR)*, 2014, pp. 2593–2600.

[5] Aaron Chadha, Alhabib Abbas, and Yiannis Andreopoulos, "Video classification with cnns: Using the codec as a spatio-temporal activity sensor," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[6] Bowen Zhang et al., "Real-time action recognition with enhanced motion vector CNNs," in *Proc. IEEE Conf. Comp. Vis. Pattern Rec. (CVPR)*, 2016, pp. 2718–2726.

[7] Alessandro Redondi, Matteo Cesana, and Marco Tagliasacchi, "Rate-accuracy optimization in visual wireless sensor networks," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*. IEEE, 2012, pp. 1105–1108.

[8] Alessandro Redondi, Luca Baroffio, Matteo Cesana, and Marco Tagliasacchi, "Compress-then-analyze vs. analyze-then-compress: Two paradigms for image analysis in visual sensor networks," in *Multimedia Signal Processing (MMSP), 2013 IEEE 15th International Workshop on*. IEEE, 2013, pp. 278–282.

[9] Karsten Suhring Tobias Oelbaum Athanasios Leontaris Alexis Tourapis, Gary Sullivan, "H.264 reference software. http://iphome.hhi.de/suehring/tml/," 2009.

[10] "Advanced video coding for generic audio-visual services, ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC)," ITU-T and ISO/IEC JTC 1, May 2003, and subsequent editions.

[11] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circ. and Syst. for Video Technol.*, vol. 13, no. 7, pp. 560–576, 2003.

[12] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[13] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *Proc. Int. Conf. Learn. Repr. (ICLR)*, 2015.

[14] Jia Deng et al., "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comp. Vis. Pattern Rec. (CVPR)*. IEEE, 2009, pp. 248–255.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1026–1034.

[16] Do-Kyoung Kwon, Mei-Yin Shen, and C-C Jay Kuo, "Rate control for H.264 video with enhanced rate and distortion models," *IEEE Trans. Circuits and Syst. for Video Technol.*, vol. 17, no. 5, pp. 517–529, 2007.

[17] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402 and CRCV-TR-12-01*, Nov. 2012.

[18] Thomas Brox and Jitendra Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *IEEE Trans. on Patt. Anal. Mach. Intel.*, vol. 33, no. 3, pp. 500–513, 2011.

[19] Karen Simonyan and Andrew Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Advances in Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 568–576.

[20] Gül Varol, Ivan Laptev, and Cordelia Schmid, "Long-term temporal convolutions for action recognition," *IEEE Trans. Patt. Anal. Mach. Intel.*, to appear.

[21] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 4489–4497.